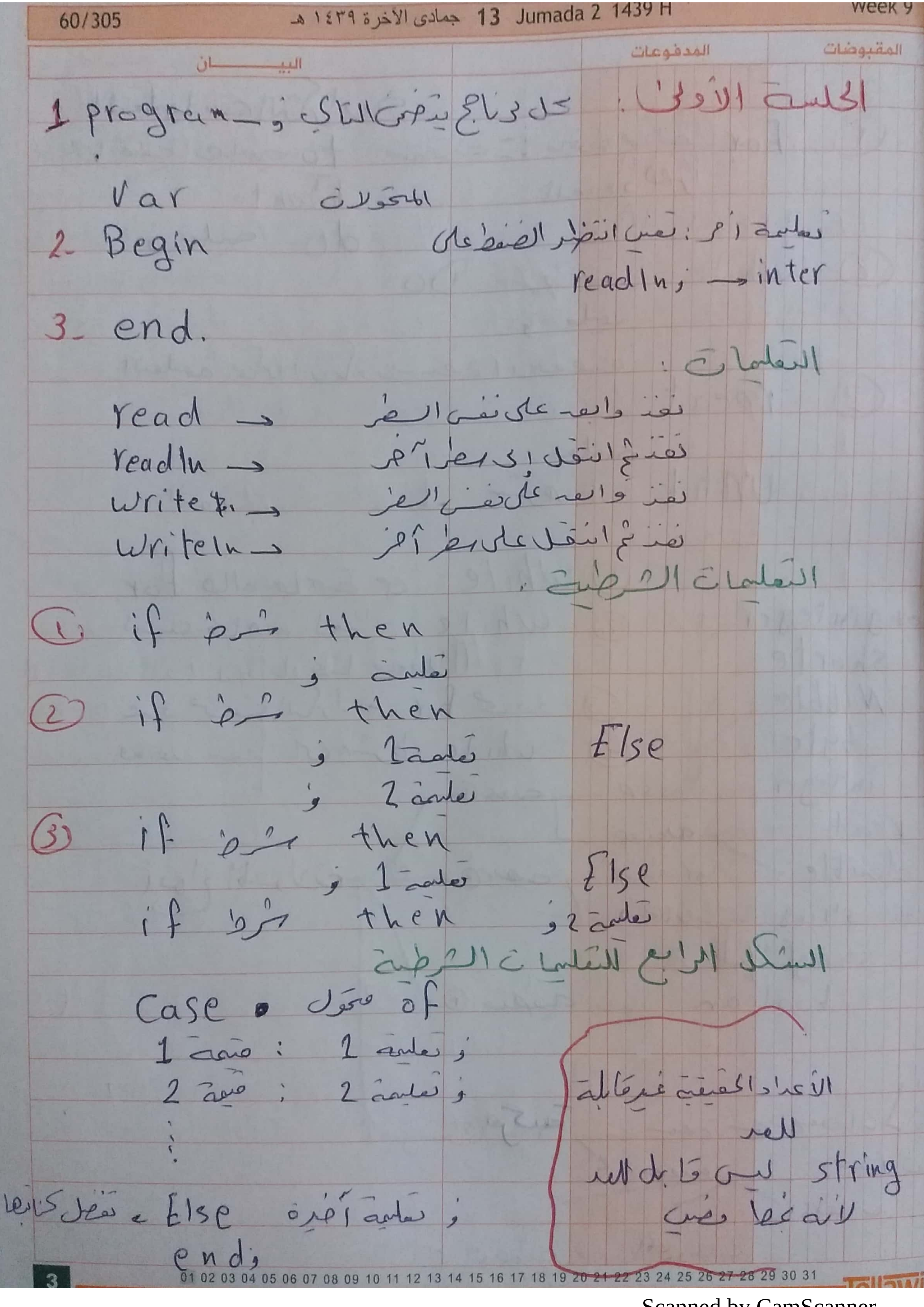


البيــــان	المدفوعات	المقبوضات

الجلسة الأولى :

كل برنامج يتضمن التالي :

1 program ـــ ;

Var المتحولات

2 Begin

تعليمة أمر: تعني انتظر الضغط على
readln; → inter

3_ end.

التعليمات :

read → نفذ واتبعه على نفس السطر
readln → نفذ ثم انتقل إلى سطر آخر
write → نفذ واتبعه على نفس السطر
writeln → نفذ ثم انتقل على سطر آخر

التعليمات الشرطية :

(1) if شرط then
تعليمة ;

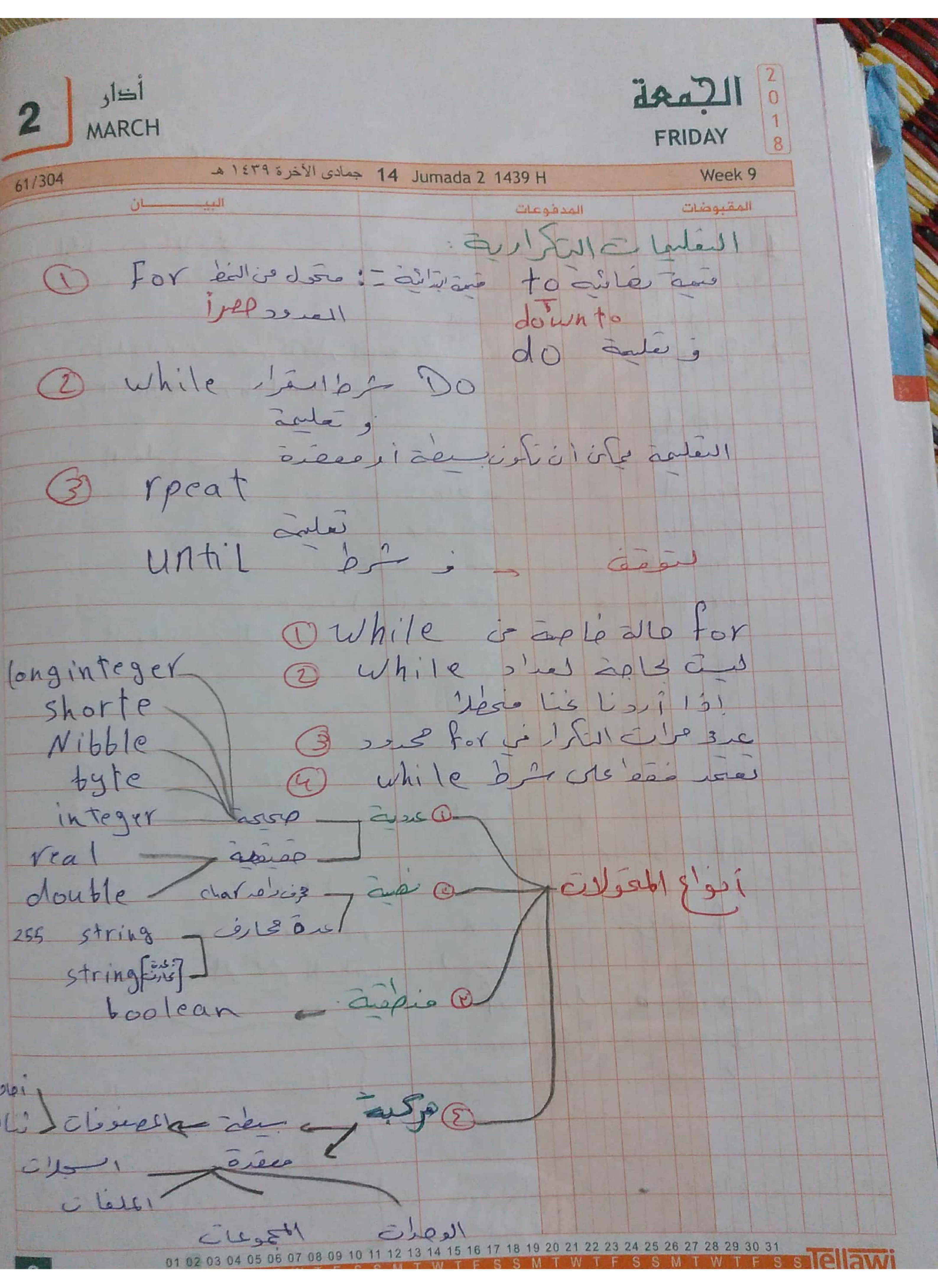
(2) if شرط then
تعليمة1 ;
Else
تعليمة 2 ;

(3) if شرط then
تعليمة 1 ;
Else
if شرط then
تعليمة 2 ;

الشكل الرابع للتعليمات الشرطية

Case متحول of
قيمة 1 : تعليمة 1 ;
قيمة 2 : تعليمة 2 ;
:
Else تعليمة أخرى ; ← تفضل كتابتها
end;

الأعداد الحقيقية غير قابلة للعد
string ليس قابل للعد لأنه نصي


2
أذار
MARCH
الجمعة
FRIDAY
2018
61/304
١٤ جمادى الأخرة ١٤٣٩ هـ
14 Jumada 2 1439 H
Week 9
البيان
المدفوعات
المقبوضات
التعليمات التكرارية:
① For متحول من النمط العدد صحيح := قيمة ابتدائية to قيمة نهائية
downto
do و تعليمة
② while شرط الاستمرار Do
و تعليمة
التعليمة يمكن أن تكون بسيطة أو معقدة
③ rpeat
تعليمة
until شرط ← للتوقف
① while حالة خاصة من for
② while ليست بحاجة لعداد
إذا أردنا عدا منتظما
③ عدد مرات التكرار في for محدود
④ while تعتمد فقط على شرط
longinteger
shorte
Nibble
byte
integer
صحيحة
real
double
حقيقية
① عددية
char حرف واحد
عدة محارف
② نصية
255 string
string[عدد المحارف]
boolean
③ منطقية
أنواع المتحولات
④ مركبة
بسيطة
المصفوفات
معقدة
السجلات
الملفات
المجموعات
الوحدات
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Tellawi

البيـــان | المدفوعات | المقبوضات

المصفوفات

Var

اسم المتحول : نوعه

X, Y, Z : integer;

a: array [دليل بداية .. دليل نهاية] of نوع العناصر

a: array [1 .. 10] of integer

char string

المصفوفات الثنائية

a: array [دليل بداية .. نهاية , بداية .. نهاية]

حالة خاصة: عندما عدد الأسطر ساوي الأعمدة تسمى مصفوفة مربعة

a_{11}	a_{12}	a_{13}
a_{21}	a_{22}	a_{23}
a_{31}	a_{32}	a_{33}

القطر الرئيسي : $i = j$

القطر الثانوي : $i + j = n + 1$

	القطر الرئيسي	القطر الثانوي
على	$i = j$	$i + j = n + 1$
فوق	$i < j$	$i + j < n + 1$
تحت	$i > j$	$i + j > n + 1$

البيـــــــان | المدفوعات | المقبوضات

المحاضرة الثالثة

برنامج لإيجاد التوافيق

$$C_n^k = \frac{n!}{k!(n-k)!}$$

```
Program AA;
Var c: real; n,k: integer;
    f1, f2, f3, i: integer;
Begin
  write('n='); readln(n);
  write('k='); readln(k);
  f1:=1;
  for i:=1 to n do
    f1:= f1 * i;
  f2:=1;
  for i:=1 to k do
    f2:= f2 * i;
  f3:=1;
  for i:=1 to (n-k) do
    f3:= f3 * i;
  c:= f1/(f2 * f3);
  writeln(c:0:2);
  readln;
end.
```

ضروري حط أقواس ضروري لأ

إذا ما حطينا أقواس يطلع 40 لأن بنفذ القسمة قبل لأن الضرب والقسمة بنفس القوة

البرامج الفرعية

readln لا يعيد قيمة إجراء بس بياخذ قيم

abs يعيد قيمة تابع بياخذ ويرجع يطبع

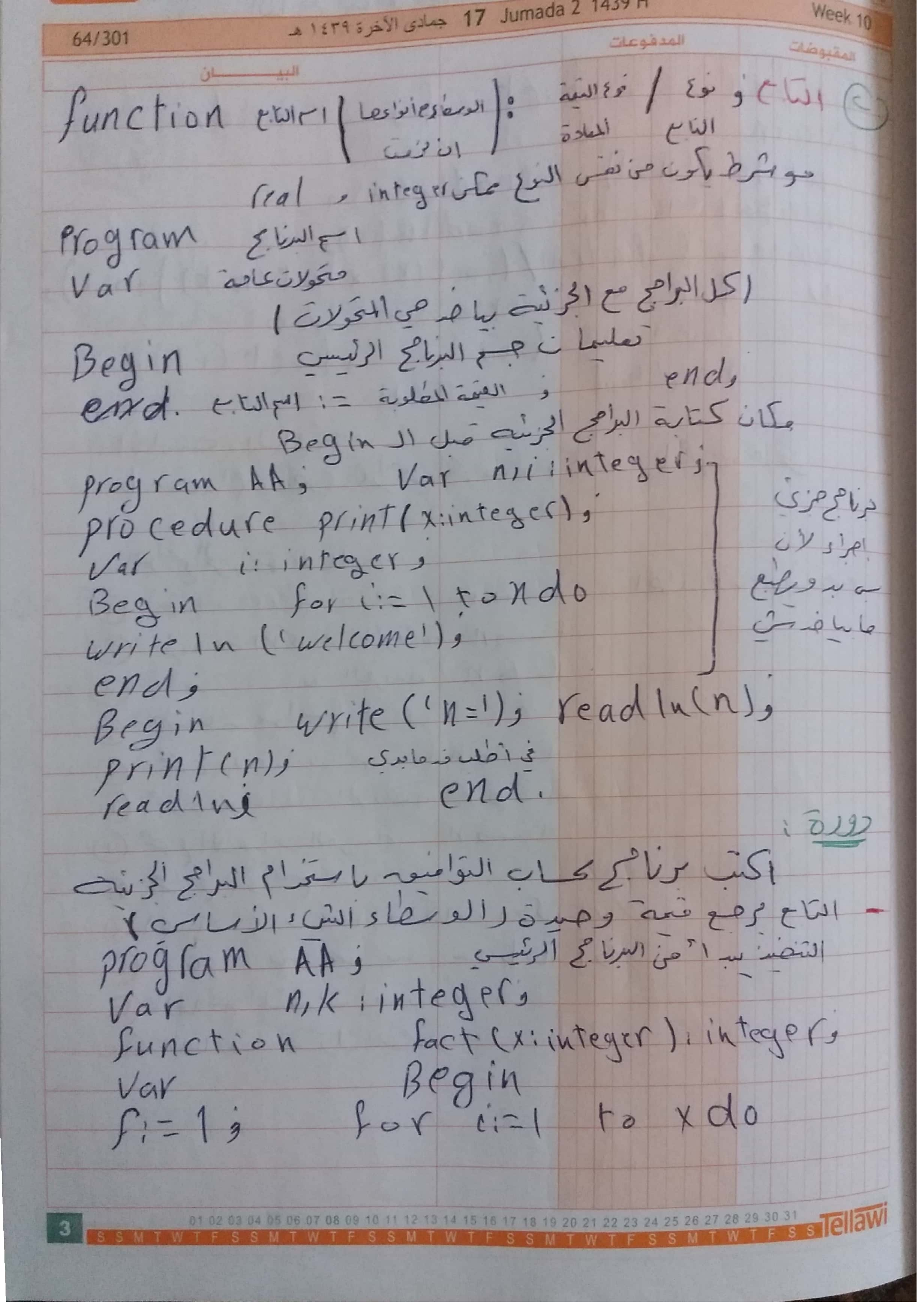
① الإجراء

procedure اسم الإجراء (أسماء الوسطاء مع أنواعها إن لزمت)

Var متحولات (خاصة بالبرنامج الجزئي)

Begin تعليمات جسم الإجراء

end;



(٢) التابع و نوع التابع / نوع القيمة المعادة

function اسم التابع (الوسطاء وأنواعها إن وجدت) :

مو شرط يكون من نفس النوع ممكن integer ، real

program اسم البرنامج
Var متحولات عامة
لكل البرامج مع الجزئية بياخد هي المتحولات !
Begin تعليمات جسم البرنامج الرئيسي
end, ; اسم التابع := القيمة المطلوبة
end.

مكان كتابة البرامج الجزئية قبل الـ Begin

```
program AA;   Var n:integer;
procedure print(x:integer);
Var   i:integer;
Begin   for i:=1 to n do
writeln('welcome');
end;
Begin   write('n='); readln(n);
print(n);
readln;
end.
```

برنامج جزئي إجرائي لأن بدو يطبع ما بياخد شي

في أطلب فما بدي

دورة :

اكتب برنامج لحساب التوافيق باستخدام البرامج الجزئية

- التابع يرجع قيمة وحيدة والوسطاء الشيء الأساسي
التنفيذ يبدأ من البرنامج الرئيسي

```
program AA;
Var   n,k:integer;
function   fact(x:integer):integer;
Var   Begin
f:=1;   for i:=1 to x do
```

```
f:= f*i;   fact:=f;   end;
Begin
  write('n='); readln(n);
  write('k='); readln(k);
  writeln( fact(n)/(fact(k)*fact(n-k)):0:2);
    for i:=1 to 10 do
    writeln(i,'  ',fact(i));
    readln;        end.
```

زيادة

سؤال

الجلسة الثالثة:

باستخدام مفهوم البرامج الجزئية اكتب برنامج يتضمن

البرامج الجزئية التالية

1) طباعة الأعداد من n → 1 — إجراء لا تابع بدون وسيط
2) طباعة الأعداد الزوجية n → 1
3) 〃 〃 الفردية n → 1
4) طباعة قواسم عدد
5) مجموع الأعداد من n → 1
6) مجموع الأعداد الزوجية n → 1
7) مجموع الأعداد الفردية n → 1
8) مجموع قواسم عدد
9) اختبار عدد فيما إذا كان أولي أو لا
10) 〃 〃 〃 〃 تاماً أو لا
11) القاسم المشترك لعددين
12) المضاعف المشترك الأصغر

البيـــان | المدفوعات | المقبوضات

```
Program AA;
Var

(1) Procedure print1(n:integer);
Var
i:integer;
Begin
        for i:=1 to n do
        writeln(i);
end;
(2) procedure print2(n:integer);
Var
i:integer
Begin   for i:=1 to n do
          if imod 2=0 then
        writeln(i);
end;
```

(3) نفسه بس الشرط if imod 2 <> 0 then
بيتغير print3 اسم

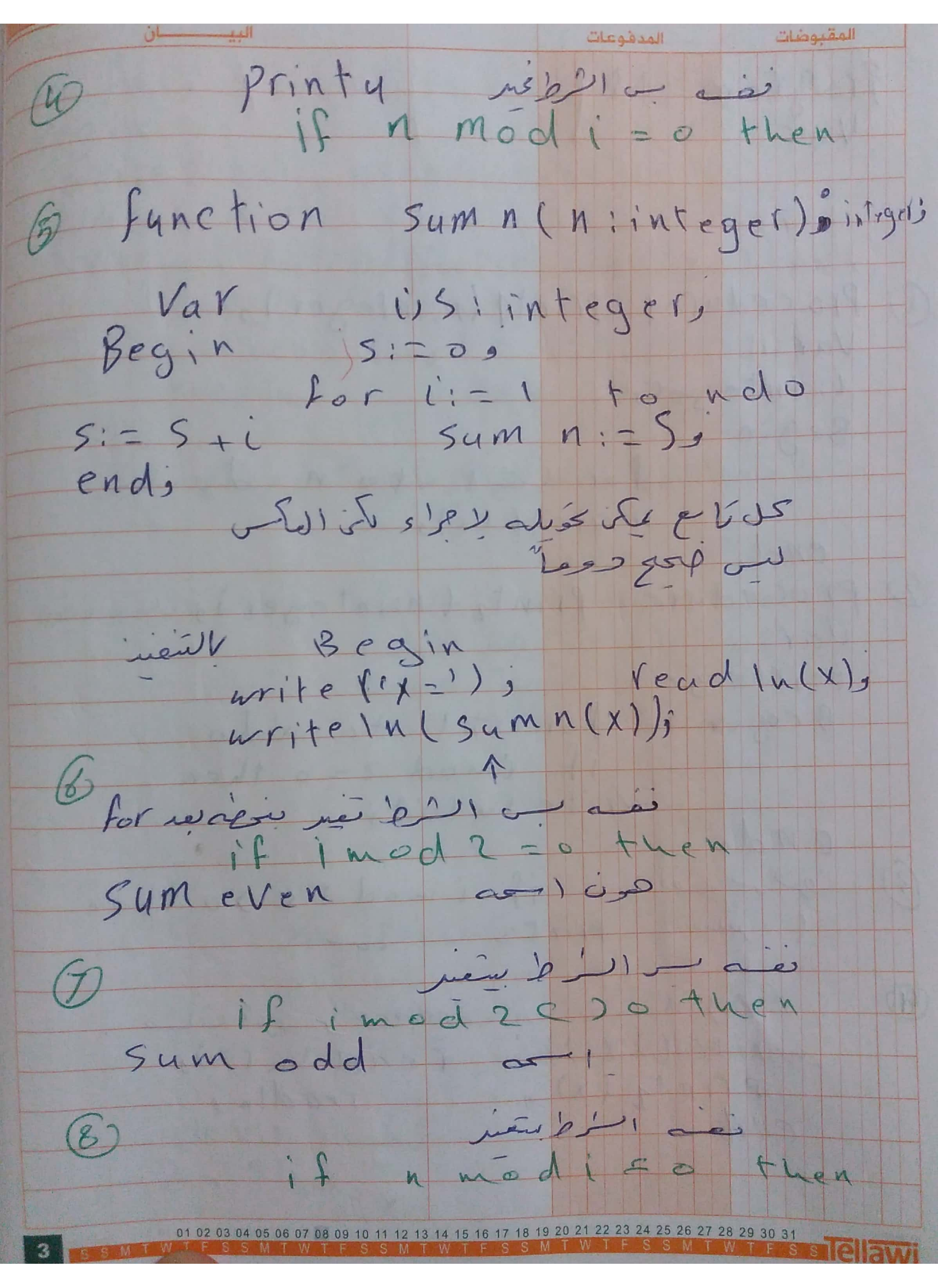
(4) مثال أعرف برامج صحيحة

```
Begin
write('x=');      readln(x);
  print1(x);         readln;
end.
```

وال لازم البرامج

يكونوا مختلفين

x, n



(4) printy نفس بس الشرط غير
if n mod i = 0 then

(5) function sum n (n : integer) : integer;

```
Var i, s : integer;
Begin s := 0;
for i := 1 to n do
s := s + i
sum n := s;
end;
```

كل تابع يمكن تحويله لإجراء لكن العكس
ليس صحيح دوماً

بالتنفيذ

```
Begin
write ('x = '); readln(x);
writeln( sum n(x));
```

(6) نفس بس الشرط تغير بنحط بعد for
if i mod 2 = 0 then
sum even غير اسمه

(7) نفس بس الشرط بيتغير
if i mod 2 <> 0 then
sum odd اسمه

(8) نفس الشرط بتغير
if n mod i ≠ 0 then

البيـــــان المدفوعات المقبوضات

Sum Q

١ ــس

السؤال الأول

```
function test prime(n:integer):boolea
Var t: boolean;
Begin      if SumQ(n) = n+1 then
test prime:=t;      t:=true
end;       else      t:= false
```

الستعانة ثم

```
if test prime(x) = true then
   writeln( x,'is prime');
write else
writeln(x,'is not prime');
readln;
end.
```

```
- Begin    readln(a);   readln(b);
for i:=a to b do
if test prime(i) = true then
writeln(i);
```

البيـــان	المدفوعات	المقبوضات

الجلسة الرابعة: متحولات البرامج العامة تحمز أصفار الجزئية لا إدخال مصفوفة إجراء

procedure inputVec(x: array[1..100] of integer;
n: integer);

↓ كلام مرفوض ما بيمشي بالباسكال

الحل: نستخدم type لبناء الأنواع أيضا بما يساويه بحالي

```
program AA;          ← يمكن التصريح عن عدة متحولات
Type      Vec = array[1..100] of integer;
Var       A: Vec;   n: integer;

Procedure  inputVec(Var x: Vec; n: integer);
```

مشان ما يتغير ((يلي بتغير عطها قدامه var مشان يتغير يحافظ على القيم الجديدة)) — سؤال

```
Var   i: integer;    Begin

for i:=1 to n do
Begin      write('x[',i,']:') do

Begin      write readln(x[i]);
end;           end;

Procedure  PrintVec(x: Vec; n: integer);
Var
i: integer;        Begin

for  i:=1  to  n do
```

البيـــان المدفوعات المقبوضات

```
write ('x[i] : 7);        writeln;
end;          Begin
write ('n='); readln (n);
  input vec (A, n);
  print vec (A, n);
  readln;
  end.
```

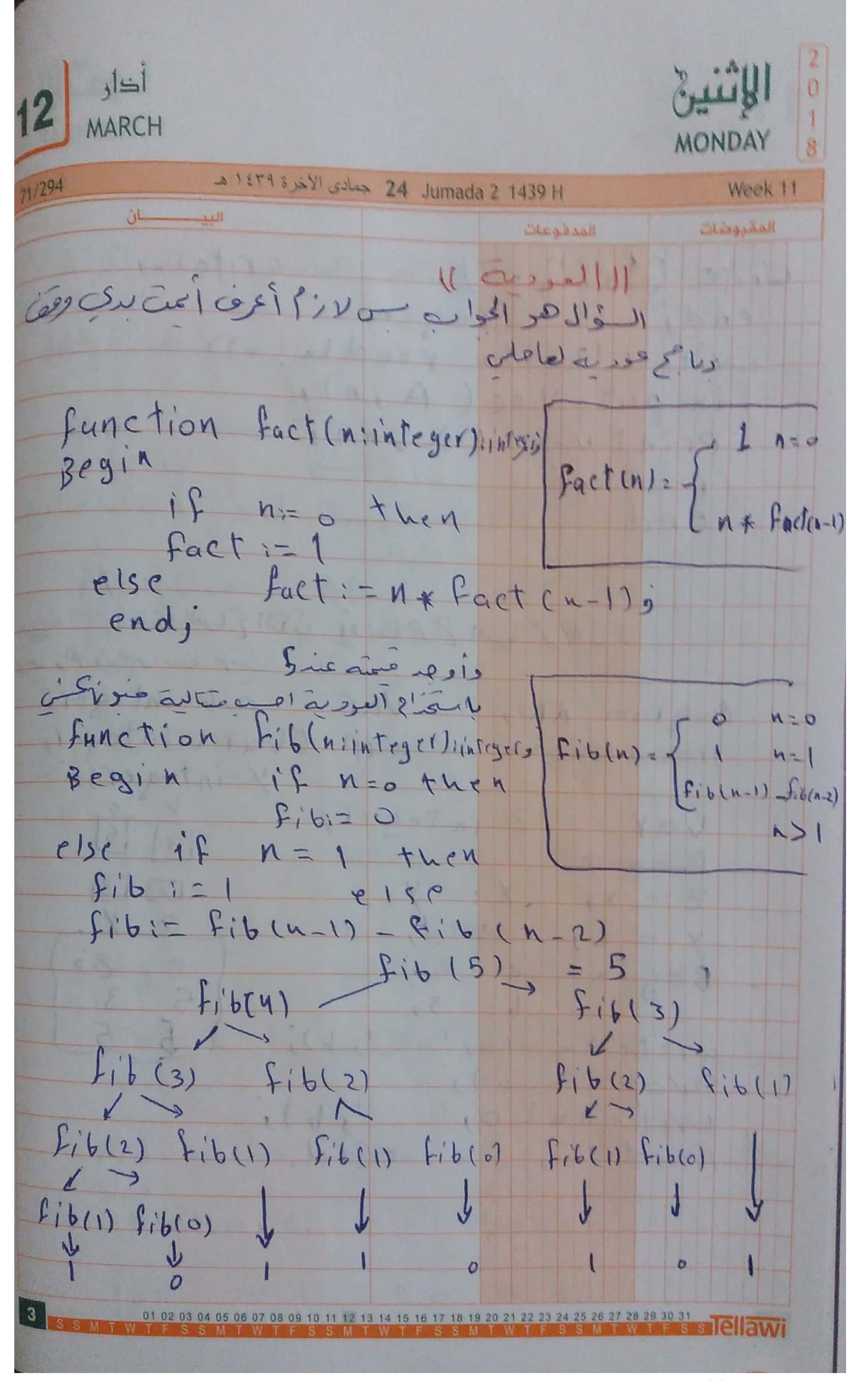
ماعمل البرنامج التالي أو ماخرج البرنامج التالي

```
program AA;     var     a,b: integer;
procedure    swap (var x: integer;
                        y: integer);
Var     z: integer;
Begin
z := x;     x := y;
y := z;     end;
Begin     a := 5;
          b := 3;
writeln (a, '     ', b);
swap (b, a);
writeln (a, '     ', b);
readln;        end.
```

b	a
3	5
3	3
z	y

خرج

a	b
5	3
5	5



البيـــــان | المدفوعات | المقبوضات

((العودية))

السؤال هو الجواب ـــ لازم أعرف أمتى بدي وقف

برامج عودية لعاملي

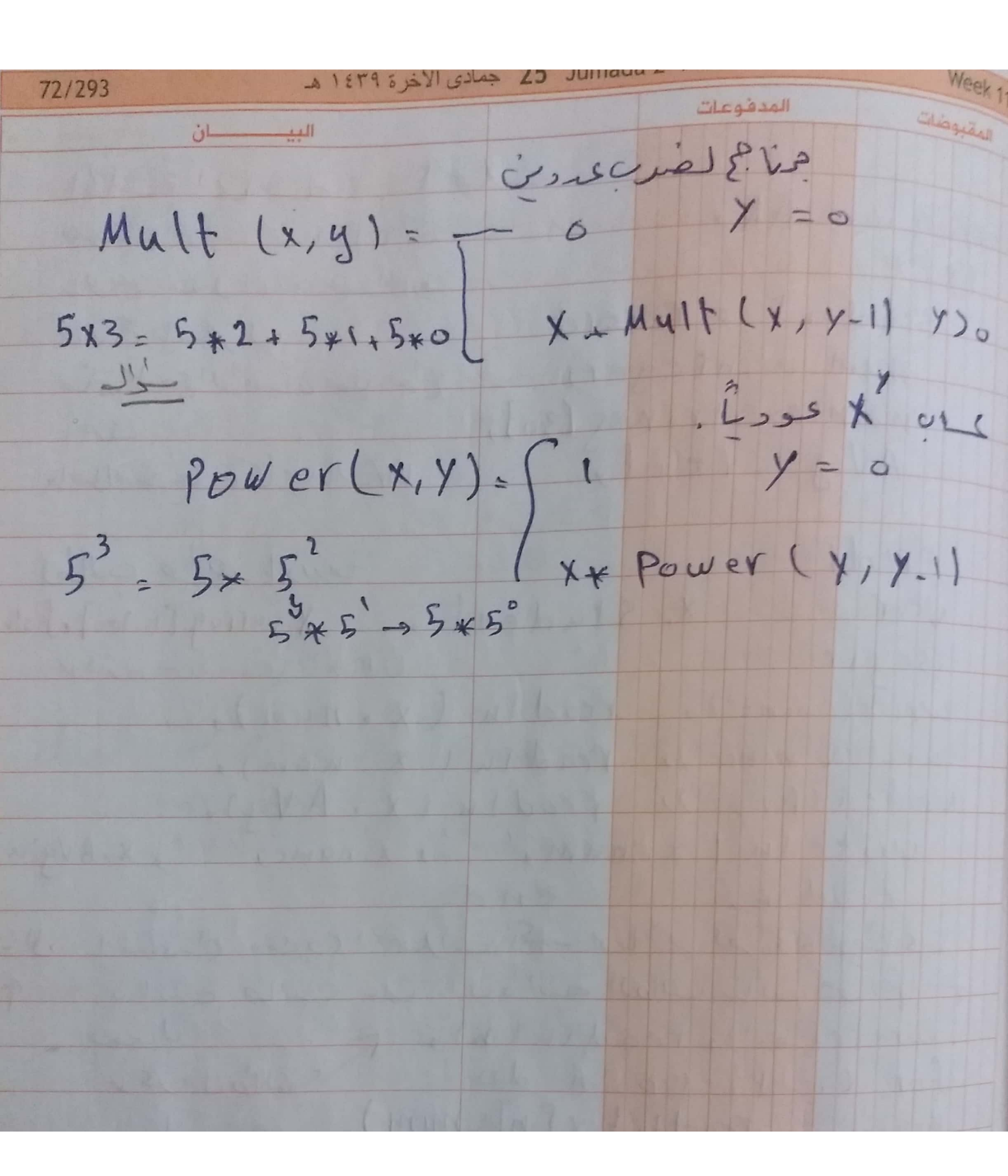
```
function fact(n:integer):integer;
Begin
      if n:= 0 then
      fact := 1
   else     fact := n * fact (n-1);
   end;
```

$$fact(n) = \begin{cases} 1 & n=0 \\ n * fact(n-1) \end{cases}$$

باستخدام العودية احسب متتالية فبوناكشي وأوجد قيمته عند 5

```
function fib(n:integer):integer;
Begin      if n=0 then
           fib:= 0
else   if  n = 1   then
   fib := 1          else
   fib:= fib(n-1) - fib (n-2)
```

$$fib(n) = \begin{cases} 0 & n=0 \\ 1 & n=1 \\ fib(n-1) - fib(n-2) & n>1 \end{cases}$$



البيـــان | المدفوعات | المقبوضات

برنامج لضرب عددين

$$\text{Mult}(x,y) = \begin{cases} 0 & y = 0 \\ x + \text{Mult}(x, y-1) & y > 0 \end{cases}$$

$5 \times 3 = 5 * 2 + 5 * 1 + 5 * 0$

سؤال

حساب x^y عودياً.

$$\text{Power}(x,y) = \begin{cases} 1 & y = 0 \\ x * \text{Power}(y, y-1) \end{cases}$$

$5^3 = 5 * 5^2$

$5 * 5^1 \rightarrow 5 * 5^0$

Scanned by CamScanner

البيـــان | المدفوعات | المقبوضات

الحلقة الخامسة: السجلات

هو نوع مركب من البيانات

بالاسم يفضل احدى عدو المعارف

تارة ع بالنظري

يمكن للنوع الواحد ان يحوي انواع أخرى

داخل التقطيعات كأنه بار Var

```
program AA;
Type
  student = record
    num : integer;
    name: string[30];
    AVg : real;
    :
  end;
  var     X: student;
  Begin
  write('num='); readln(x.num);
  write('name='); readln(x.nam);
  write('AVg='); readln(x.AVg);
  writeln(x.num,' ', x.name,' ', x.AVg:0:2);
  readln;       end.
```

لطالب واحد بالازرق

X : array [1..100] of student;

لأكثر من طالب بنجوم i, n : integer

السؤال: باستخدام مصفوفة السجلات اكتب برنامج لقراءة وطباعة ذاتية طالب علماً أن ذاتية الطالب تتألف من رقم، اسم، معدل

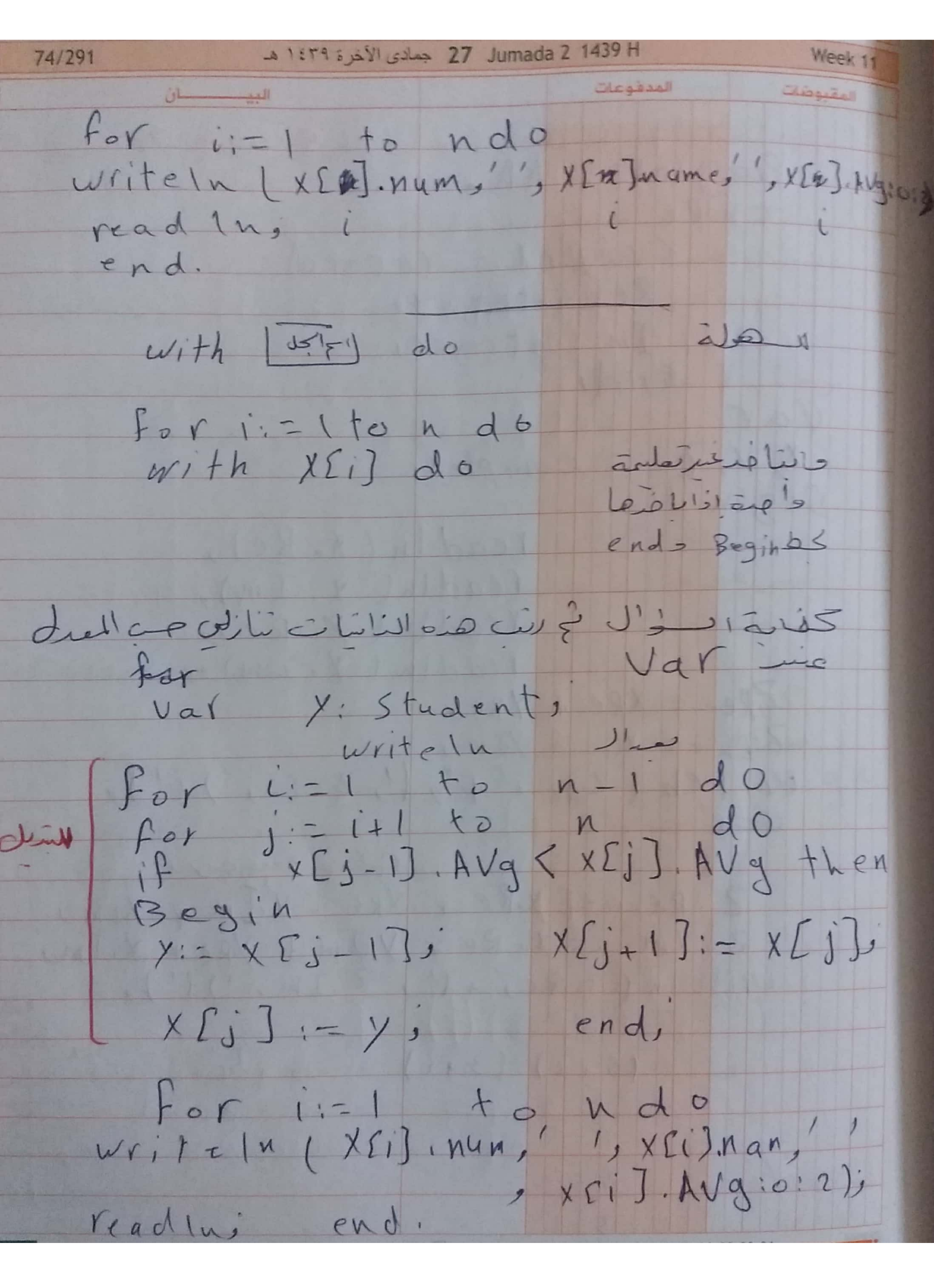
```
write('n='); readln(n);
for i:=1 to n do
Begin   readln(x[i].num);
        readln(x[i].name);
        readln(x[i].AVg);
end;   ~~writeln(x[i].num, ' ', x[i]~~
```

لأكثر من طالب

ملاحظة

X[i]. Mark[j] → مصفوفة ثانية لعلامات الطلاب

← مصفوفة طلاب



البيان   المدفوعات   المقبوضات

```
for i:=1 to n do
writeln ( x[i].num,' ', x[i].name,' ', x[i].Avg:0:2);
readln;
end.
```

```
with [اسم السجل] do
```

ملاحظة

```
for i:=1 to n do
with x[i] do
```

وانا قد غير تعليمة واحدة إذا أكثر ها كلها Begin و end

كتابة سؤال في رتب هذه البيانات تنازلي حسب المعدل

عند Var

```
for
var   y: Student;
      writeln
```

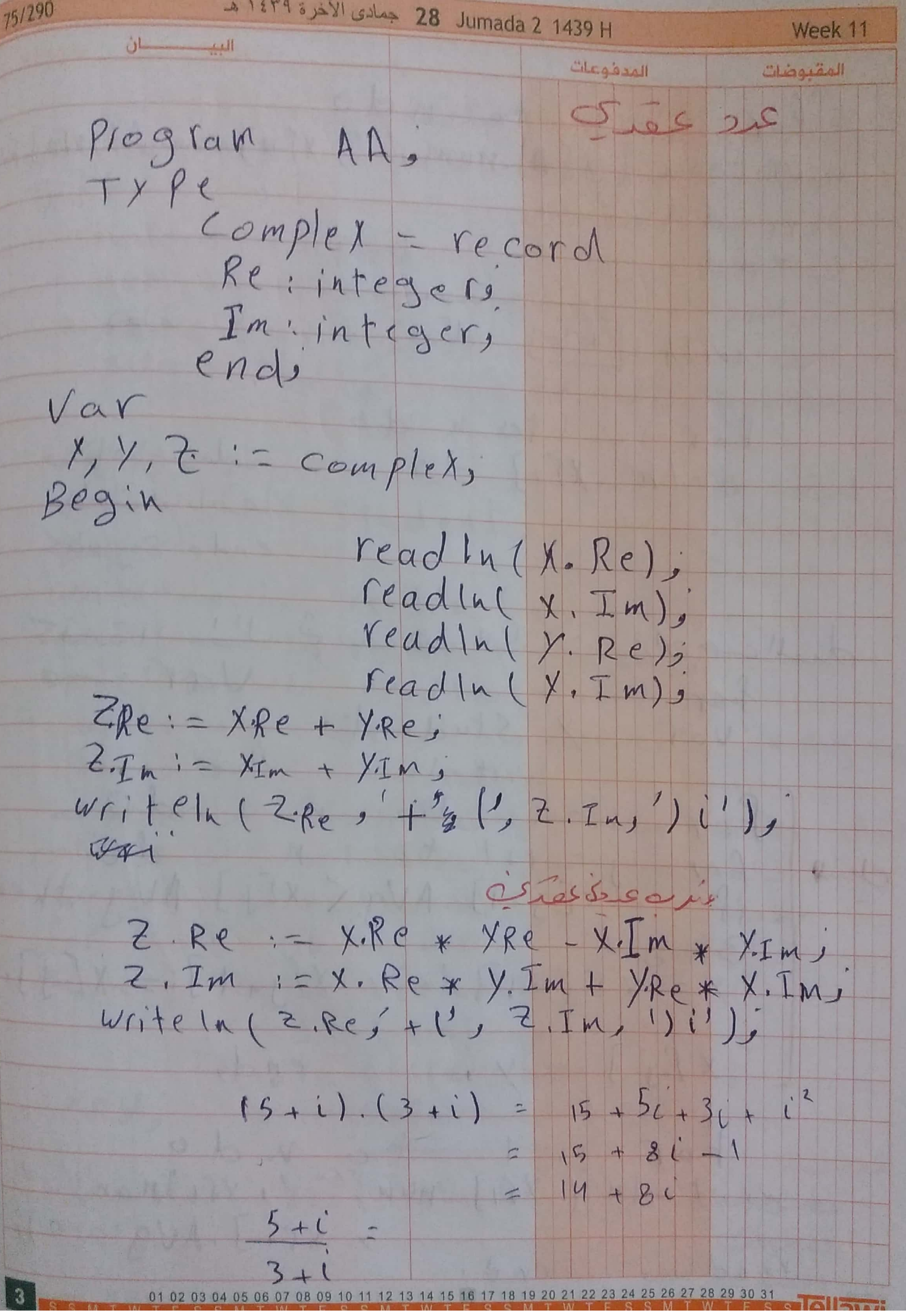
مدار

التبديل

```
for i:=1 to n-1 do
for j:=i+1 to n do
if x[j-1].AVg < x[j].AVg then
Begin
 y:= x[j-1];        x[j+1]:= x[j];

 x[j]:= y;          end;

for i:=1 to n do
writeln ( x[i].num,' ', x[i].nan,' '
                     , x[i].AVg:0:2);
readln;      end.
```



البيـــــان | المدفوعات | المقبوضات

عدد عقدي

```
Program AA;
Type
    complex = record
      Re : integer;
      Im : integer;
    end;
Var
  X, Y, Z := complex;
Begin
          readln(X.Re);
          readln(X.Im);
          readln(Y.Re);
          readln(Y.Im);
  Z.Re := X.Re + Y.Re;
  Z.Im := X.Im + Y.Im;
  writeln(Z.Re, '+(', Z.Im, ')i');
```

ضرب عددين عقديين

```
  Z.Re := X.Re * Y.Re - X.Im * Y.Im;
  Z.Im := X.Re * Y.Im + Y.Re * X.Im;
  writeln(Z.Re, '+(', Z.Im, ')i');
```

$$(5+i).(3+i) = 15 + 5i + 3i + i^2$$
$$= 15 + 8i - 1$$
$$= 14 + 8i$$

$$\frac{5+i}{3+i} =$$

إذا اردي قسم عددين عقديين نضرب البسط والمقام بمرافقه المقام

الضرب البسط بالبسط والمقام بالمقام
القسمة الضرب بالمقلوب
بالجمع والطرح نفس الشكل يلي قبلو بس عند العمليات

سؤال

```
ضرب  Z.Re := X.Re * Y.Im + Y.Re * X.Im;
     Z.Im := X.Im * Y.Im;
```

الحلقة السادسة ، الملف

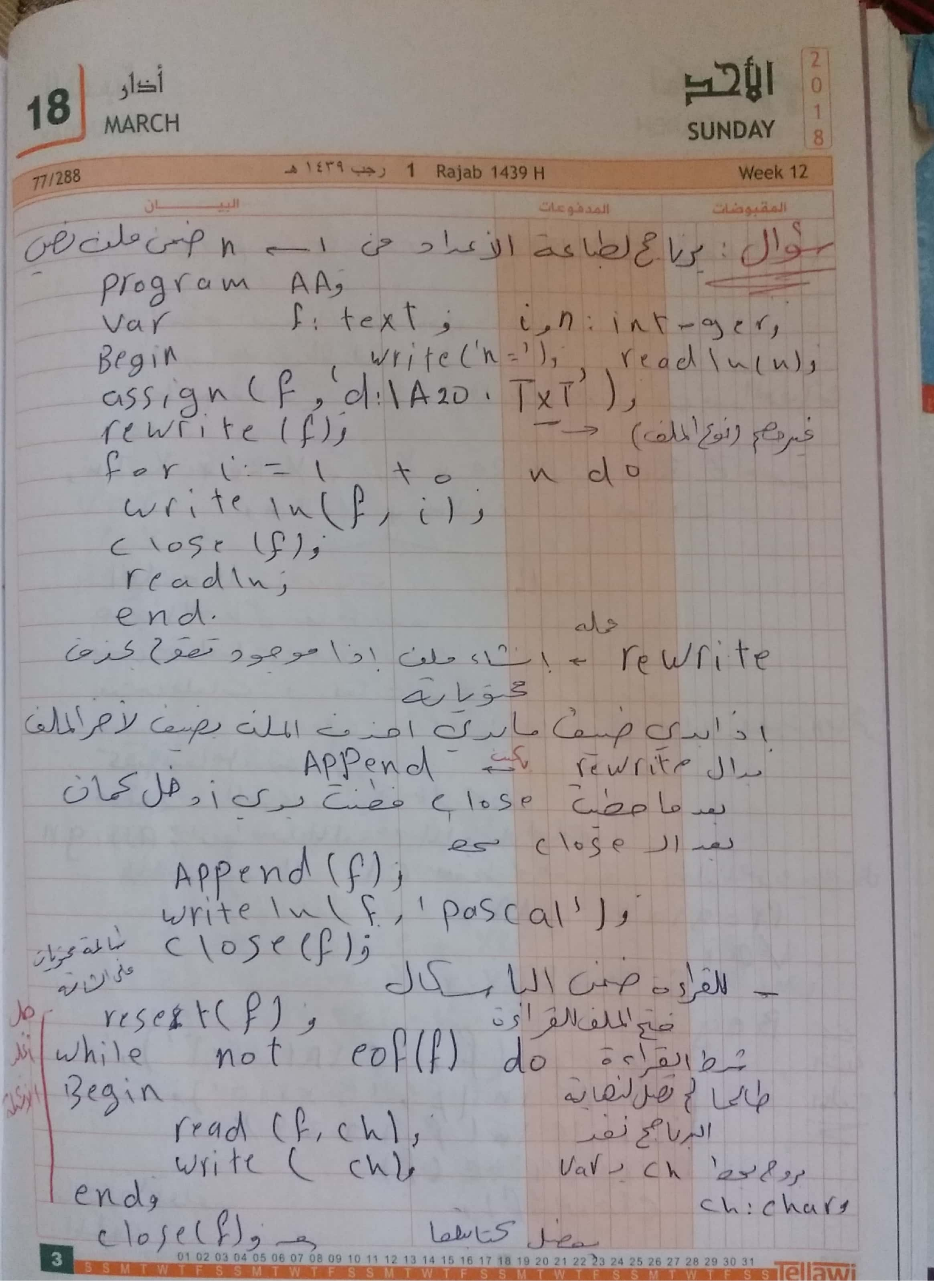
هو مثل البرنامج نتعامل معه
يسمى متحول ملف أو ملف حوفت
نوعين ملفات ١- ملفات نصية
٢- ملف ثنائي (تفرعي) لا يمكن الوصول لأي عنصر بشكل مباشر

كيف نتعامل معه
هو نوع من المتحولات بذكر بال Var
assign : لربط متحول الملف مع الملف الأصلي
إذا ما حطيت ال C:\ بحفظ لحالو بنفس المجلد الموجود فيه بالاسكال

```
Program AA;
Var
   f: text;
Begin
   assign(f, 'C:\A1.TxT');
   writeln(f, 'Syria');
   writeln(f, 1234);
   rewrite(f);
   close(f);
```

كي إنشاء الملف

ما يحفظ الإغلاق الملف



سؤال: برنامج لطباعة الأعداد من ١ ــ n ضمن ملف نص

```
program AA;
var     f: text ;    i,n: integer;
Begin      write('n=');   readln(n);
assign(f, 'd:\A20.TxT');
rewrite(f);
for i:=1 to n do
 writeln(f, i);
close(f);
readln;
end.
```

rewrite(f); ← ضروري (نوع الملف)

rewrite ← إنشاء ملف إذا موجود تحقق بحذف محتوياته

إذا بدي ضيف ما بدي امسح الملف بضيف لآخر الملف
بدال rewrite بكتب Append
بعد ما حطيت close فحطيت بدي أزود فل كمان
بعد ال close صح

```
Append(f);
writeln(f, 'pascal');
close(f);
```

شالة محتويات على الشاشة

ـ للقراءة ضمن الباسكال

```
reset(f);
while not eof(f) do
Begin
   read(f, ch);
   write( ch);
end;
close(f);
```

reset(f); ← فتح الملف للقراءة
while not eof(f) do ← شرط القراءة طالما لم تصل للنهاية
read(f, ch); ← البرنامج نفذ
write(ch); ← مرح نحط Var ب ch — ch: char;
close(f); ← مفصل كتابتها

حل أخذ الأشياء (في الهامش)

البيان المدفوعات المقبوضات

نسخ محتويات ملف لآخر

```
assign(g,'B25.TXT');

rewrite(g);
reset(f);
while not eof(f) do
Begin
 read(f,ch);
 write(g,ch);
 end;
 close(f);          close(g);
 readln;
 end.
```

استبدال كل A بـ t

لا يمكن تعديل الملف النصي ← الحل نسخه على ملف جديد

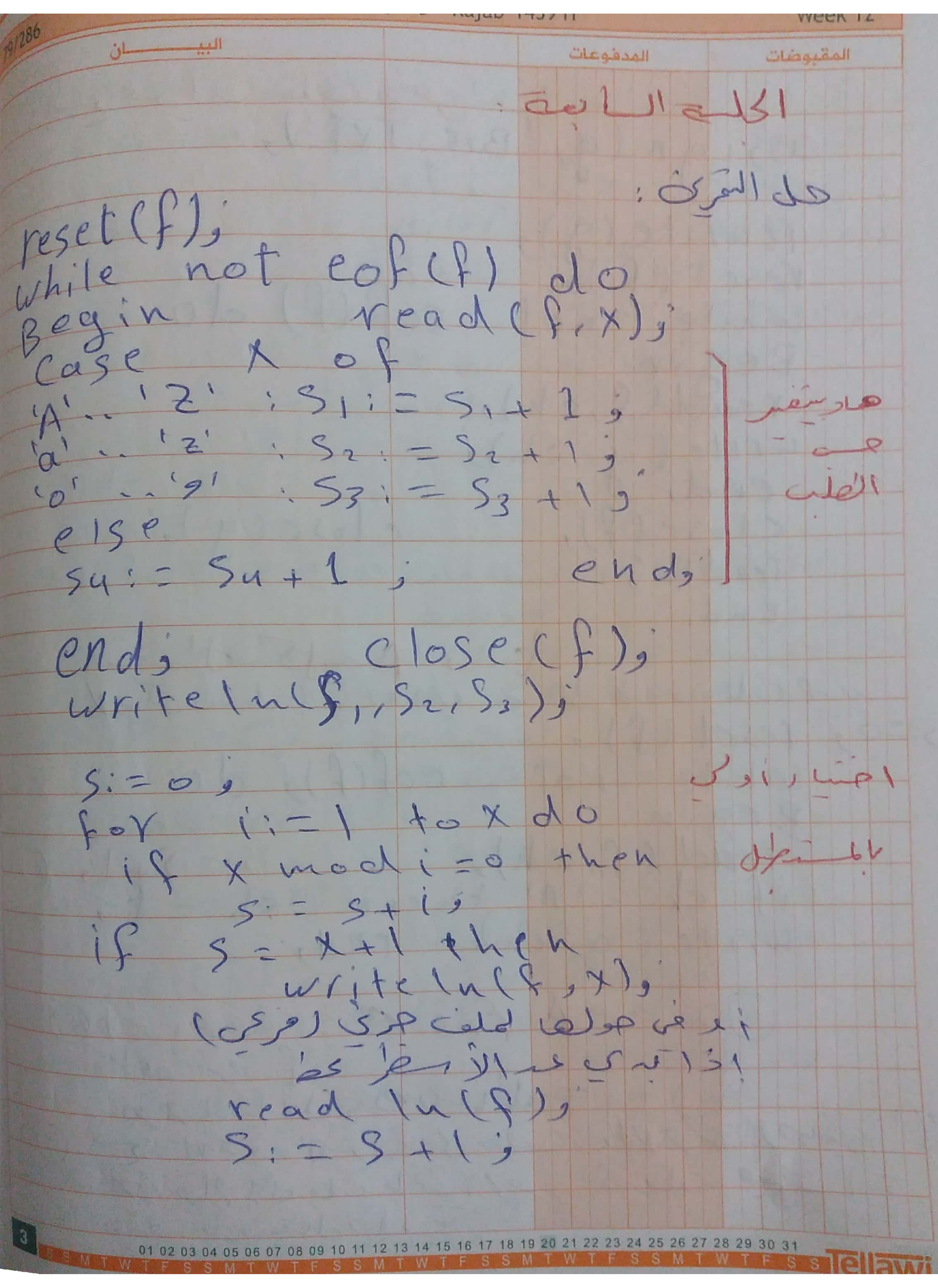
```
s:=0; reset(f);
 while not eof(f) do
 Begin
  read(f,ch);                     s:=s+1; (1)
  if ch ='A' then         ch:=t;
  write(g,ch);     end;
```

(2) عد أحرف - عدد A موجودة (1)

ما حط شرط if

تعرض لدينا ملف نص المطلوب:

(1) نسخ الأحرف الكبيرة إلى ملف والأحرف الصغيرة إلى ملف الأرقام إلى ملف باقي الرموز إلى ملف مع حساب عدد أحرف كل ملف



المقبوضات	المدفوعات	البيـــان

الحلقة السابعة :

حل التمرين :

```
reset(f);
while not eof(f) do
Begin      read(f,x);
 case   x  of
 'A'..'Z'  : s1:= s1+1;
 'a'..'z'  : s2:= s2+1;
 '0'..'9'  : s3:= s3+1;
  else
  s4:= s4+1 ;          end;

end;       close(f);
 writeln(f,s1,s2,s3);
```

هاد بيتغير حسب الطلب

```
 s:=0;
 for  i:=1  to x do
   if  x mod i=0  then
        s:= s+i;
  if   s = x+1 then
         writeln(f,x);
```

احتمال اولى

بالتسلسل

أي في حولها لملف جزئي (مركزي)

إذا بدي عد الأسطر كلها

```
     readln(f);
     s:= s+1;
```

الملفات الثنائية

```
x: array[1..100] of Vec
```

مصفوفة عرّفنا سابقاً

file ← شكل الثنائي
أي كل عنصر ضمن المصفوفة هو مصفوفة

العناصر متجانسة يفضل أخذ الملف الثنائي إذا كله ثنائي

يمكن تحويل الثنائي لنص ولكن العكس غير ممكن

```
Var
f: file of integer;

Begin
assign(f, 'c:\A:txt');
rewrite(f);
write('x='); readln(x);
write(f, x);
close(f);
reset(f);
```

تسمح للتعديل والإضافة والقراءة

ممنوع استخدام الـ ln بالثنائي مثل writeln readln

بالثنائي ما ضروري نعلم الملف close(f)

بالنص كل حرف يأخذ بايت واحد 111 بايت
أما بالثنائي 0 .. 32767 ← كله 2 بايت

الثنائي ما في تعليمة Append

الـ reset ← تسمح للتعديل والإضافة والقراءة لا يمكن تعديل النص

كفاءة البرنامج

```
reset(f);
while not eof(f) do
  Begin
  read(f,x);
  {
  end;            close(f);
```

بالنص عا ستخدم for
الثاني نفس

```
reset(f);
n:= file size(f);
for i:=1 to n do
  Begin
        read(f,x);

  end;        close(f);
```

باستخدام for

ملاحظة: ب read نستعملها اذا كانت القراءة للوحة المفاتيح عادي مع n

اكتب برنامج لإنشاء ملف من الأعداد الصحيحة علماً أن عددها n عدد ثم أوجد عدد الأعداد الزوجية في هذا الملف

```
Program AA;
  Var
f2,f1,f:file of integer;
   n,i,x,S:integer;
```

الأعداد فانها متتالية عشوائية

```
Begin
assign(f,'C:\A10.TXT');
rewrite(f);
write('n='); readln(n);
for i:=1 to n do
Begin
write('x='); readln(x);
write(f,x);
end;
close(f);
s:=0;
reset(f);
while not eof(f) do
Begin
read(f,x);
if x mod 2 <> 0 then
s:=s+1;
end;
close(f);
writeln('s=',s);
readln;
end.
```

إضافة ①

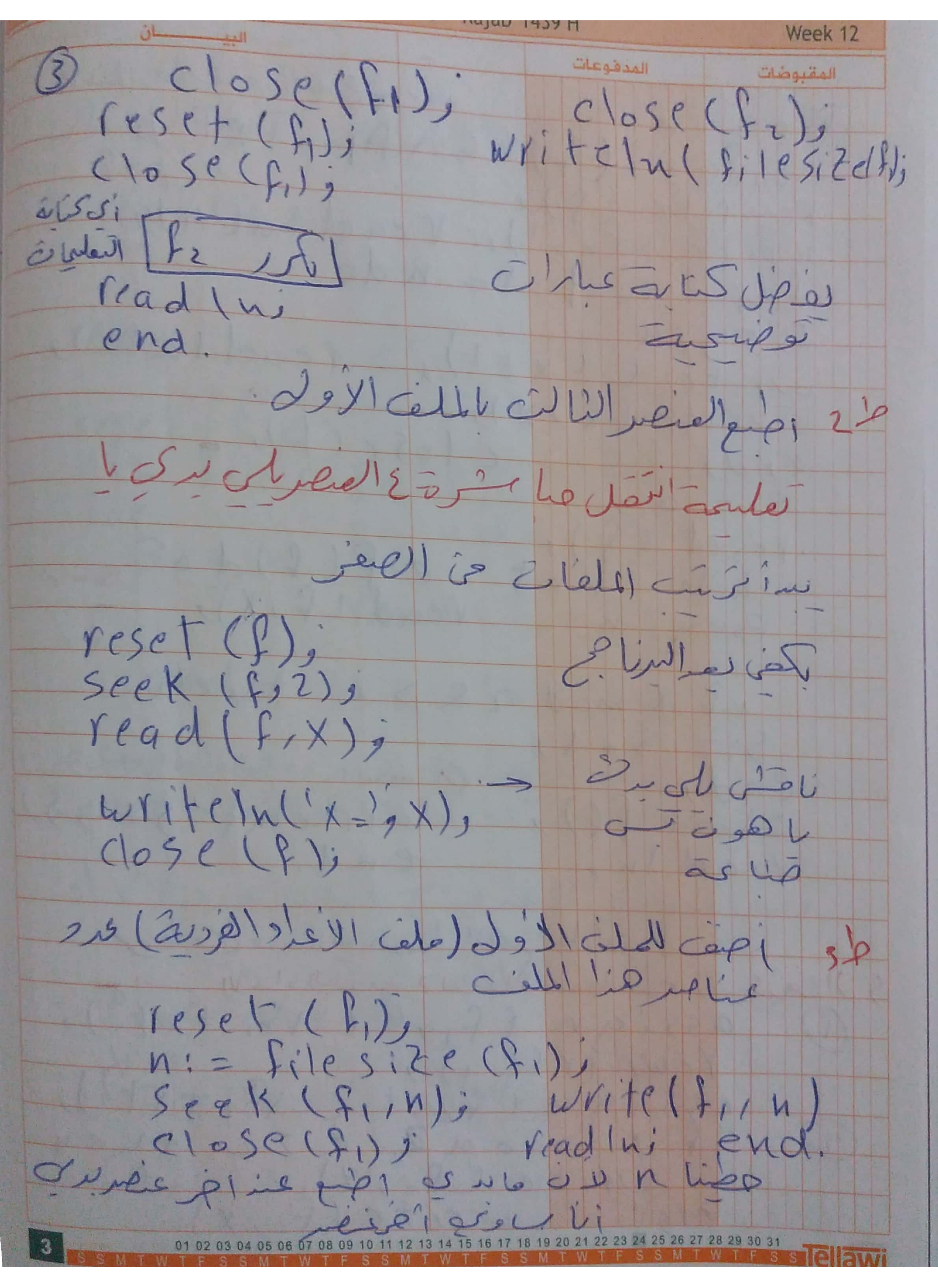
إضافة ②

إضافة ③

الربط دون مابين مصفوفة الصم قبل reset

- انسخ الأعداد الفردية لملف والزوجية لملف ... حذف الـ

```
① assign(f1,'c:\f1.txt');
  rewrite(f1);
  assign(f2,'c:\f2.txt');
② if x mod 2 <> 0 then
      write(f1,x)
   else write(f2,x)
```



③
```
close(f1);
reset(f1);
close(f1);
```
أي كتابة التعليمات

f2 تكرار

```
readln;
end.
```

```
close(f2);
writeln(filesize(f1));
```

يفضل كتابة عبارات توضيحية

ط2) اطبع العنصر الثالث بالملف الأول

ملاحظة انتقل مباشرة ع العنصر يلي بدي يا

بدأ ترتيب الملفات من الصفر

```
reset(f);
seek(f,2);
read(f,x);

writeln('x=',x);
close(f);
```

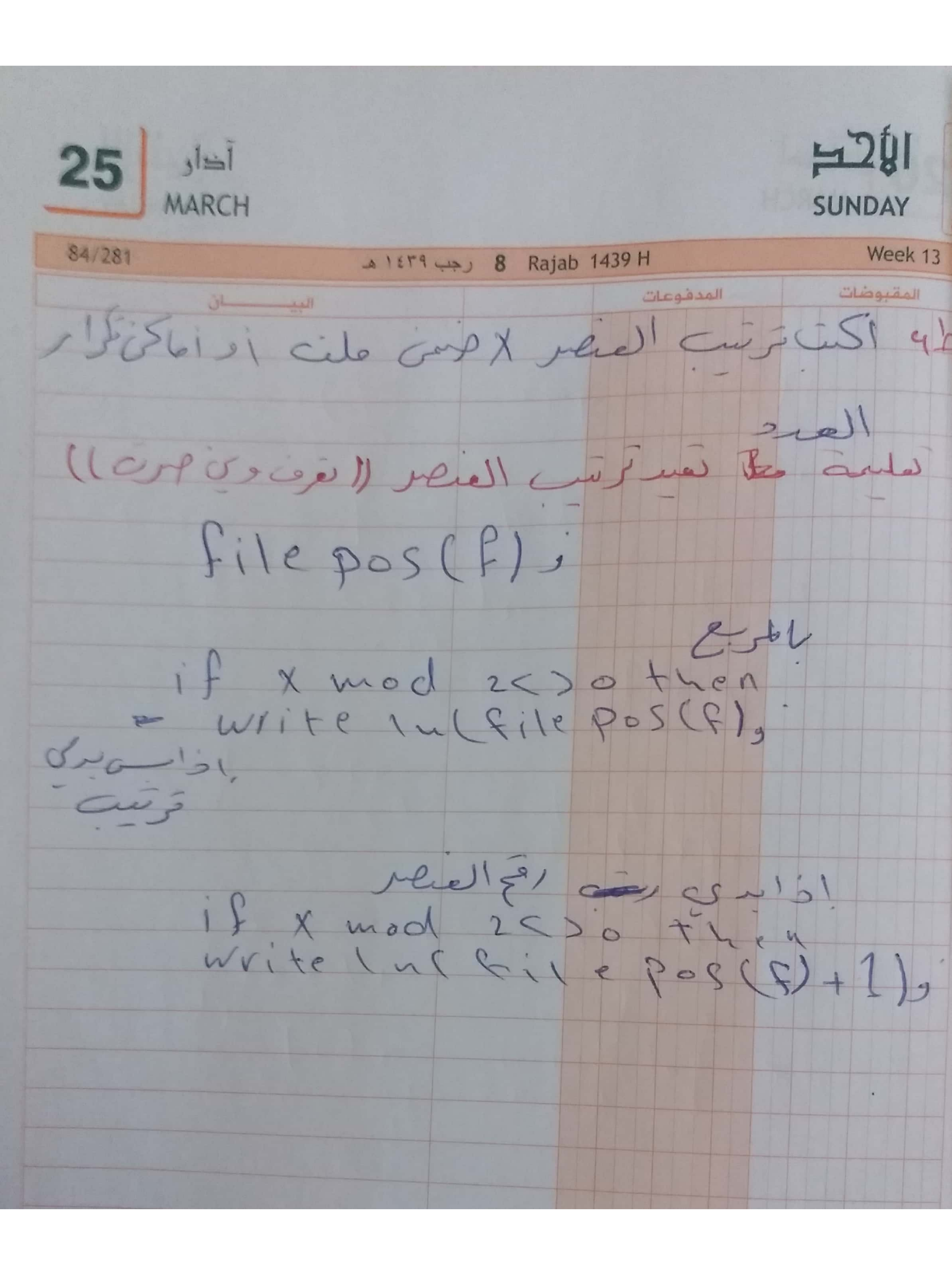
يكفي نص البرنامج

ناقص يلي بدي باهون تبع طباعة ←

ط3) أضف للملف الأول (ملف الأعداد الفردية) عدد عناصر هذا الملف

```
reset(f1);
n:=filesize(f1);
seek(f1,n);      write(f1,n)
close(f1);       readln;   end.
```

حطينا n لأن ما بدي اضع عند اخر عنصر بدي انا بدي بعد اخر عنصر



ط٤/ أكتب ترتيب العنصر X ضمن ملف أو أماكن تكرار العدد

تعليمة تفيد ترتيب العنصر (التعرف على مرتين)

```
file pos(f);
```

بالمربع

```
if x mod 2<>0 then
  write ln(file pos(f));
```

إذا بدي ترتيب

إذا بدي رقم العنصر

```
if x mod 2<>0 then
write ln(file pos(f)+1);
```